

Amendments to the Claims

The listing of claims will replace all prior versions, and listings of claims in the application.

1. *(currently amended)* A computer-implemented method for interfacing between one or more requestors and one or more airline availability information sources, comprising the steps of:

(1) receiving a first request from a first requestor for airline availability information;

(2) querying one or more airline availability information sources for the requested airline availability information;

(3) receiving the requested airline availability information from the one or more airline availability information sources;

(4) caching the received airline availability information;

(5) providing the received airline availability information to the requestor;

(6) receiving a second query from a second requestor for the airline availability information; [and]

(7) determining ~~to provide~~ amongst providing the second requestor with at least ~~one of~~ the following types of airline availability information:

real-time information; and

cached information; and

(8) providing information to the second requestor in accordance with the determination made in step (7).

2. *(previously presented)* The method according to claim 1, further comprising the steps of:

(9) monitoring airline availability information traffic between an airline availability information source and one or more clients of the airline availability information source; and

(10) caching at least a portion of the monitored airline availability information.

3. *(previously presented)* The method according to claim 1, further comprising the steps of:

(9) proactively generating one or more queries independent of requestor queries; and

(10) sending the one or more proactively generated queries to an airline availability information source and caching information returned therefrom.

4. *(previously presented)* The method according to claim 1, further comprising the steps of:

(9) monitoring airline availability information traffic between an airline availability information source and one or more clients of the airline availability information source;

(10) caching at least a portion of the monitored airline availability information;

(11) proactively generating one or more queries independent of requestor queries; and

(12) sending the one or more proactively generated queries to an airline availability information source and caching information returned therefrom.

5. (*original*) The method according to claim 3, further comprising the steps of:

(11) adding the requestor queries to a query priority queue;

(12) adding proactively generated queries to the query priority queue, at lower priorities than the requestor queries; and

(13) processing the requestor queries and the proactively generated queries according to their priorities.

6. (*previously presented*) The method according to claim 5, wherein step (11) comprises the steps of:

(a) separating a first requestor query into one or more sub-queries;

(b) prioritizing the one or more first requestor sub-queries with respect to one another;

(c) placing the one or more first requestor sub-queries in the query priority queue;

(d) separating a second requestor query into one or more sub-queries;

(e) prioritizing the one or more second requestor sub-queries with respect to one another; and

(f) placing the one or more second client sub-queries in the query priority queue, ordering the first requestor sub-queries with respect to the second requestor sub-queries according to associated times of receipt, resolving priority disputes between simultaneously received first and second requestor queries so that higher priority sub-queries of the first and second requestors are processed before lower priority sub-queries of the first and second requestors.

7. (*previously presented*) The method according to claim 3, wherein step (9) comprises the step of proactively generating queries to populate cache.

8. (*previously presented*) The method according to claim 3, wherein step (9) comprises the step of proactively generating queries to update cached information.

9. (*original*) The method according to claim 3, wherein step (9) comprises the step of ordering the proactive queries for processing based on time-to-departures and age of associated cached information.

10. (*original*) The method according to claim 9, wherein step (9) further comprises the steps of:

- (a) generating a plurality of storage buckets in a memory;
- (b) associating at least a portion of the buckets with various time-to-departures;
- (c) ordering the buckets according at least to their associated time-of-departures;

(d) bucketing the proactive queries according at least to their associated time-to-departures;

(e) ordering the proactive queries within the buckets at least according to ages of previously cached data associated with the proactive queries;

(f) re-bucketing the proactive queries as their associated time-to-departures change; and

(g) selecting a bucket for processing according to the ordering of step (9)(c), processing the proactive queries within the selected bucket, skipping proactive queries for which information is presently cached and newer than a predetermined age.

11. (*original*) The method according to claim 10, wherein:

step (9)(b) comprises the step of associating the buckets with various time-to-departures and according to one or more modes of transportation, and

step (9)(c) comprises the step of ordering the buckets according to the nearness to time-of-departures and the associated modes of transportation.

12. (*original*) The method according to claim 10, further comprising the steps of:

(11) adding the requestor queries to a query priority queue;

(12) adding proactively generated queries from buckets selected in accordance with step (9)(g), to the query priority queue, at lower priorities than the requestor queries; and

(13) processing the requestor queries and the proactively generated queries in the query priority queue according to their priorities.

13. (*previously presented*) The method according to claim 12, wherein step (11) comprises the steps of:

- (a) separating a first requestor query into one or more sub-queries;
- (b) prioritizing the one or more first requestor sub-queries with respect to one another;
- (c) placing the one or more first requestor sub-queries in a query priority queue;
- (d) separating a second requestor query into one or more sub-queries;
- (e) prioritizing the one or more second requestor sub-queries with respect to one another; and
- (f) placing the one or more second client sub-queries in the query priority queue, ordering the first requestor sub-queries with respect to the second requestor sub-queries according to associated times of receipt, resolving priority disputes between simultaneously received first and second requestor queries so that higher priority sub-queries of the first and second requestor are processed before lower priority sub-queries of the first and second requestor.

14. (*original*) The method according to claim 1, wherein:

step (6) comprises the step of receiving a second requestor preference for real-time information and/or cached information; and

step (7) comprises the step of determining to provide the second requestor with real-time information and/or cached information based at least in part on the second requestor preference.

15. *(previously presented)* The method according to claim 1, wherein step (8) comprises the step of determining to provide the second requestor with real-time information and/or cached information based at least in part on one or more of the following factors:

- an availability of requested information in cache;
- a currently cached flight availability count;
- a client preference for cached and/or realtime data;
- an age of the cached information;
- a client identification and/or client importance factor;
- a time of day;
- a proxy availability;
- availability of recently cached information;
- one or more rules associated with an information source;
- an activity/load at a realtime information source;
- anticipated turn around time to an information source;
- total number of seats;
- a nearness to time-to-departure;
- a market importance;
- a frequency of prior availability changes; and
- a mode of transportation.

16. *(previously presented)* The method according to claim 1, further comprising the step of:

(9) querying one or more information sources through one or more proxies.

17. (*previously presented*) The method according to claim 16, wherein step (9) comprises the step of:

(a) monitoring an operational status of the one or more proxies and selecting proxies for querying based on the monitored operational status.

18. (*previously presented*) The method according to claim 16, wherein step (9) comprises the step of:

(a) monitoring response times for the one or more proxies and selecting proxies for querying based at least on the response times.

19. (*previously presented*) The method according to claim 16, wherein step (9) comprises the step of:

(a) maintaining a list of unsupported suppliers for which information is not available on the one or more information sources; and

(b) returning queries for information from the unsupported suppliers without querying an information source.

20. (*original*) The method according to claim 16, wherein step (9) comprises the steps of:

(a) maintaining proxy records for available proxies in a proxy queue;

(b) removing a higher priority proxy record from the proxy queue to process a query.

21. (*original*) The method according to claim 20, wherein step (9)(a) comprises the steps of maintaining the proxy queue as part of a query priority queue.

22. (*previously presented*) The method according to claim 1, wherein step (1) further comprises the step of receiving a first request from a first requestor for one or more of the following additional types of information:

hotel availability information;
rental car availability information;
taxi availability information;
entertainment availability information; and
restaurant availability information;

wherein steps (2) through (8) are performed with the one or more types of additional types of information.

23. (*original*) The method according to claim 14, wherein step (6) further comprises the step of permitting the second requestor to select one of the following options:

return real-time data only;
return cached data only;
return cached data if available, otherwise consult real-time data source;
and

return cached data if the cached data is less than N seconds old, otherwise consult real-time data source.

24. (*original*) The method according to claim 14, wherein step (6) further comprises the step of permitting the second requestor to select and prioritize a plurality of the following options:

return real-time data only;

return cached data only;

return cached data if available, otherwise consult real-time data source;

and

return cached data if the cached data is less than N seconds old, otherwise consult real-time data source.

25. (*original*) The method according to claim 1, further comprising the step of:

(9) caching recently updated information separately from less recently updated information and searching the recently updated cached information when real-time data is sought.

26. (*original*) The method according to claim 1, further comprising the steps of:

(9) permitting the requestors to specify approximate departure times in the requests for airline availability information; and

(10) searching a cache for requested information.

27. (*previously presented*) The method according to claim 26, wherein step (4) comprises the steps of:

(a) rounding-up actual departure times for each flight, providing at least the rounded-up actual departure times to a hashing function, and storing information associated with the flights in a hash table based on resulting rounded-up hash table indexes; and

(b) rounding-down actual departure times for each flight, providing at least the rounded-down actual departure times to the hashing function, and storing information associated with the flights in the hash table based on resulting rounded-down hash table indexes;

wherein step (10) comprises the steps of:

(a) rounding-up a user specified departure time, providing the rounded-up user specified departure time to the hash function, and searching the hash table based on a resulting hash table index; and

(b) rounding-down a user specified departure time, providing the rounded-down user specified departure time to the hash function, and searching the hash table based on a resulting hash table index.

28. (*original*) The method according to claim 1, further comprising the steps of:

(9) initiating a control thread for a request, whereby the request includes one or more sub-queries;

(10) initiating a worker thread for each sub-query associated with the request;

(11) prioritizing the worker threads with respect to one another; and

(12) processing the worker threads according to associated priorities.

29. (*original*) The method according to claim 1, wherein step (4) comprises the steps of sharing a flight availability count record between a plurality of flight records stored in the cache.

30. (*previously presented*) The method according to claim 1, wherein step (4) comprises the steps of:

(a) associating multiple flight records as married flight records in the cache; and

(b) sharing a flight availability count record between at least one of the multiple flight records and another flight record in the cache.

31. (*previously presented*) The method according to claim 1, wherein step (8) comprises the step of searching for cached information after waiting a pre-determined time for real-time information.

32. (*original*) The method according to claim 1, further comprising the step of:

(9) communicating with the one or more information sources through proxies, whereby the proxies interface with one or more of the information sources using information source specific codes.

33. (*original*) The method according to claim 32, wherein step (9) further comprises the steps of:

- (a) measuring one or more response characteristics associated with the proxies;
- (b) prioritizing the proxies according to the performance measurements; and
- (c) maintaining a proxy priority queue, whereby queries are passed to higher priority proxies.

34. (*original*) The method according to claim 32, wherein step (9) further comprises the steps of:

- (a) identifying one or more information sources that proxies cannot communicate with; and
- (b) filtering out queries directed to the identified information sources.

35. (*original*) The method according to claim 32, wherein step (9) further comprises the steps of:

- (a) monitoring an operational status of the proxies; and
- (b) optimizing use of the proxies based on the operational status of the proxies.

36. (*original*) The method according to claim 32, further comprising the step of:

(10) simulating replies from the proxies.

37. (*original*) The method according to claim 3, wherein step (10) comprises the step of sending the one or more proactively generated queries periods of low information source activity.

38. (*original*) The method according to claim 3, wherein step (9) comprises the step of generating background threads that pose queries that appear to come from requestors.

39. (*original*) The method according to claim 3, wherein step (9) comprises the step of filtering one or more queries out of proactive caching.

40. (*original*) The method according to claim 39, wherein step (9) further comprises the step of filtering out queries related to airline flights for which fares are not available.

41. (*original*) The method according to claim 39, wherein step (9) further comprises the step of filtering out queries related to flights on unsupported carriers.

42. (*original*) The method according to claim 39, wherein step (9) further comprises the step of filtering out queries related to flights that users are not expected to request.

43. (original) The method according to claim 3, wherein step (9) comprises the step of assigning priority to queries according to an associated market.

44. (original) The method according to claim 3, wherein step (9) comprises the step of assigning priorities to queries according to a frequency of flights.

45. (original) The method according to claim 3, wherein step (9) comprises the step of assigning priorities to queries according to a frequency of changes associated with availability of corresponding flights.

46. (original) The method according to claim 3, wherein step (9) comprises the step of assigning priority to queries according to a market importance.

47. (original) The method according to claim 3, wherein step (9) comprises the step of assigning priority to queries according to nearness of departure time.

48. (original) The method according to claim 3, wherein step (9) comprises the step of assigning priority to queries according to an age of cached data.

49. (original) The method according to claim 3, wherein step (9) comprises the step of assigning priority to queries according to a number of remaining available seats.

50. (original) The method according to claim 3, wherein step (9) comprises the step of assigning priority to queries according to anticipated increases in travel volume.

51. (original) The method according to claim 3, wherein step (9) comprises the step of assigning priority to queries according to a type of product/service.

52. (original) The method according to claim 3, wherein step (9) further comprises the step of assigning lower priority to forms of ground transportation.

53. (original) The method according to claim 3, wherein step (9) further comprises the step of assigning lower priority to flights that use propeller planes.

54. (original) The method according to claim 42, wherein step (2) further comprises the step of assigning priority according to a total number of available seats.

55. (original) The method according to claim 3, wherein step (9) comprises the step of updating cached airline availability information according to multiple priorities.

56. (original) The method according to claim 55, wherein step (9) further comprises the step of encoding the multiple priorities into a mathematical function that

assigns a combined priority value to units of airline availability information, and updating the airline availability information according to the associated combined priority values.

57. (*original*) The method according to claim 55, wherein step (9) further comprises the steps of:

(a) prioritizing airline availability information according to departure times;

(b) prioritizing airline availability information according to one or more additional features; and

(c) updating the airline availability information based on a combination of the priorities associated with the departure time and one or more additional features.

58. (*original*) The method according to claim 1, further comprising the step of:

(9) predicting an availability status.

59. (*original*) The method according to claim 58, wherein step (9) comprises the step of predicting availability status based on prior observed variables, including prior availability information.

60. (*previously presented*) The method according to claim 59, wherein step (9) further comprises the steps of:

- (a) identifying one or more factors associated with availability status;
- (b) learning a relationship between historical values for the one or more factors and historical values for availability status;
- (c) generating a function according to the learned relationship; and
- (d) providing new values for the one or more factors to the function, whereby the function outputs predicted values for availability status.

61. (*original*) The method according to claim 1, further comprising the steps of:

- (9) separating a first requestor query into one or more sub-queries;
- (10) prioritizing the one or more first requestor sub-queries with respect to one another;
- (11) placing the one or more first requestor sub-queries in a query priority queue;
- (12) separating a second requestor query into one or more sub-queries;
- (13) prioritizing the one or more second requestor sub-queries with respect to one another;
- (14) placing the one or more second client sub-queries in the query priority queue, ordering the first requestor sub-queries with respect to the second requestor sub-queries according to associated times of receipt, resolving priority disputes between simultaneously received first and second requestor queries so that higher priority sub-queries of the first and second requestors are processed before lower priority sub-queries of the first and second requestors; and

(15) processing the queries in the query priority queue according to their associated priorities.

62. (*previously presented*) The method according to claim 1, further comprising the steps of:

(9) monitoring airline availability information traffic between an airline availability information source and one or more clients of the airline availability information source;

(10) determining a likelihood that information will be received in a near future by said monitoring;

(11) generating proactive queries for information not likely to be received in the near future; and

(12) caching information returned in response to the proactive queries.

63. (*currently amended*) A computer-implemented method for interfacing between one or more requestors and one or more information sources, comprising the steps of:

(1) receiving a first request from a first requestor for information;

(2) querying one or more information sources for the requested information;

(3) receiving the requested information from the one or more information sources;

(4) caching the received information;

- (5) providing the received information to the requestor;
- (6) receiving a second query from a second requestor for the information;
- (7) determining ~~to provide~~ amongst providing the second requestor with at least ~~one of~~ the following types of information:
 - real-time information; and
 - cached information; and
- (8) providing information to the second requestor in accordance with the determination made in step (7).

64. (*currently amended*) A computer program product including a computer useable medium having computer program logic stored therein to enable a computer to interface between one or more requestors and one or more information sources, wherein said computer program logic comprises:

a receiving function that causes the computer system to receive requests for information from information requestors;

a query process function that causes the computer system to determine ~~whether to process~~ amongst processing a query, at least, out-of-cache [or] and with real-time information, and that causes the computer system to query one or more information sources when it determines to process a query with real-time information; and

a cache control function that causes the computer system to cache information returned from the one or more information sources.